Smart Computing Review

The Sybil Attacks and Defenses: A Survey

Aziz Mohaisen[1] and Joongheon Kim[2]

[1] Verisign Labs, Verisign Inc. / 12061 Bluemont way, Reston, VA 20151, USA / amohaisen@verisign.com

[2] University of Southern California / Los Angeles, CA, USA / joongheon.kim@usc.edu

*Corresponding Author: Aziz Mohaisen

Received September 23, 2013; Revised November 30, 2013; Accepted December 7, 2013; Published December 19, 2013

Abstract: In this paper we take a close look at the Sybil attack and advances in defending against it, with particular emphasis on recent work. We identify three major veins in the research literature that describe ways to defend against the attack: using trusted certification, using resource testing, and using social networks. The first vein in the literature considers defending against the attack using trusted certification, which is done by either centralized certification or distributed certification using cryptographic primitives that can replace the centralized certification entity. The second vein in the literature considers defending against the attack by testing resources, which can be in the form of IP testing, network coordinates, or recurring cost (e.g., by requiring clients to solve puzzles). The third and last vein in the literature is by mitigating the attack, combining social networks used as bootstrap security and tools from random walk theory, which was shown to be effective in defending against the attack under certain assumptions. Our survey and analyses of the different schemes in the three veins in the literature show several shortcomings, which form several interesting directions and research questions worthy of investigation.

Keywords: Sybil attacks and defenses, reputation, distributed systems

Introduction

The peer-to-peer (P2P) paradigm of computing has a lot of advantages over other conventional paradigms. For example, in this paradigm, resources such as bandwidth, memory, and data are made available to other all participating users [1]. Broadly, this paradigm includes structured and unstructured systems. Structured overlays, such as Kademlia [2] and Chord [3], provide deterministic mechanisms for data and peer discovery, whereas unstructured overlays, such as Gnutella [4], organize peers in a random graph and use flooding for peers and data discovery. Most of the popular peer-to-peer systems lack centralized authorities, which makes this paradigm robust against failure attacks. On the other hand, the lack of such centralized authorities leads to many challenging security issues; most services necessary for securing networked systems require one type of centralized authority or another, making these services unavailable to peer-

DOI: 10.6029/smartcr.2013.06.009

to-peer systems [5]. Even worse, the fully decentralized and open nature of many of these systems enables a wide range of security threats unknown in other distributed systems, including the Sybil attack [6].

The Sybil attack (so named from the 1973 book on multiple personalities by Flora Rheta Schreiber) is well known in the context of peer-to-peer, wired, and wireless networks. In its basic form, a peer representing the attacker generates as many identities as she can and acts as if she is multiple peers in the system [6] aiming at disturbing the normal behavior of the system. The number of identities that an attacker can generate depends solely on the attacker's capabilities, which are limited by the bandwidth required for responding to concurrent requests by other peers in the system, the memory required for storing routing information of other peers corresponding to each and every generated Sybil identity, and computation resources required for serving concurrent requests without noticeable delay. With sharp hardware growth (e.g., in terms of storage capacity and processing power) as well as the spread of broadband Internet with high bandwidth rates, even attackers running on "commodity" hardware can cause substantial harm to large systems.

The attack itself is popular and effective in many contexts and on services that are essential in peer-to-peer systems as well as other generic distributed systems and paradigms. Such contexts include voting systems, reputation systems, routing, and distributed storage, among others. To illustrate how this attack works in real systems, imagine a recommender system built over a peer-to-peer overlay [7]. In such a system, the goal is to filter information that is likely to be of interest to users based on others' recommendations. In that context, an attacker who can act as multiple users by faking multiple identities can easily out-vote legitimate users' votes on legitimate objects that are subjected to voting. This is almost guaranteed, given that the number of legitimate users who normally vote is always no more than 1% of the total number of users in any realistic recommendation system [7]. Such an attack becomes appealing to potential users trying to take advantage of a system that provides incentives. For example, many online marketplaces, such as eBay, use recommendations from customers to determine the reputation of the people who use the platform to sell goods, and thus there is an incentive for such sellers to gain a better reputation. The same scenario arises in many other contexts, such as peer-to-peer file sharing where content is rated by users, where bandwidth is assigned based on reputation, or when reputation is used to determine the value of content distributed by users. In all such examples, users have an incentive to take unfair advantage, and the Sybil attack has proven a powerful tool for attackers to achieve such goals.

To defend against the attack, there have been several attempts in the form of defenses, or mitigations, to defend against, or limit, the impact of the attack. Such attempts can be classified broadly into two schools of thoughts: centralized and decentralized (i.e., distributed) defenses. In centralized defenses [6, 8-10], a centralized authority is responsible for verifying the identity of each and every user in the system. While this defense is somewhat effective in defending against the attack, it makes certain assumptions about the system, some of which are not easy to achieve in peer-to-peer decentralized systems. First of all, as the name and description implies, such systems require a centralized authority, which might not be affordable for both security and functionality reasons. Even if such a centralized authority exists, it requires credentials for users in the system to match against each user's digital identity. In many settings, obtaining such credentials is very challenging.

On the other hand, many decentralized defenses [7, 11-23] do not require such authorities and are well designed for decentralized peer-to-peer systems. At the core of their operation, such defenses weigh collaboration among users in the system to admit or reject users who are potential attackers. Admission or rejection of users is based on credentials associated with them, as in the case of cryptographic distributed defenses, or network properties of legitimate, honest users, as in the case of Sybil defenses using social graphs. In either of these solutions, the ultimate goal of the defense is to simulate the power of the centralized authority in a decentralized manner and use such power to detect both Sybil and honest nodes.

Another classification of defense could be according to the way the defense operates. Accordingly, existing defenses in the literature can be classified into those using 1) trusted certification—in which certificates are typically generated for honest users and verified against a public key of a trusted authority, 2) incurring cost—in which users are penalized in a way that limits their available resources and thus reduces any misbehavior, and 3) social network–based Sybil defenses.

These defenses differ greatly in their assumptions, in the type of network they are applied to, in the guarantees they provide, and in the costs incurred. To this end, this paper reviews, summarizes, compares, and shows shortcomings in the existing literature on such defenses. Two aspects characterize our method in this survey: first, we review each category of defense and show their merits in defending against the attack in the claimed context. Second, we summarize the direction by showing the main shortcomings that lead to open problems worthy of investigation. The latter part sheds light on the technical contributions towards solving the problem that are fragile and that require a lot of investigation in order to solve the problem.

To summarize the rest of this paper, in Section 2 we introduce preliminaries, including a functional classification of defenses. In Section 3 we review the trusted certification–based category of defense, while in Section 4 we introduce the category of resource testing, followed by social network–based defenses in Section 5. In Section 6, we discuss related work, followed by concluding remarks in Section 7.

Model, Settings, and Objectives

In this section, we elaborate on the problem at hand and describe the attacker model conventionally assumed by most Sybil attack studies and defenses. We further explore the objectives of the attacker and the objective of the defenses proposed in the literature.

Problem Statement and Model

The problem is the ability of a single user in the system to act as if he is multiple users with different identities. This is problematic for many applications where correctness depends on the behavior of peers, their numbers, and their willingness to collaborate honestly. However, such a goal cannot be satisfied when Sybil identities of a single attacker try to bias the overall behavior of the system.

The number of fake identities that can be injected into the system formally characterizes the attacker. The attacker's objective is to maximize this number. The value and meaning of the number of identities generated by the attacker and injected into the system depends on the application itself and varies from application to application. For example, to attack a recommender system, it is enough to match with fake identities just 1% of the honest users in the system. This is enough to bias the behavior of the system and outvote the honest nodes, because it is generally accepted that, even for the most popular objects, only 1% of the honest nodes vote for them. Thus, having an identity that generates more than the number of honest nodes voting on an object enables the Sybil attacker to outvote the honest nodes.

In other systems, such as mixing networks used for communication anonymization (e.g., the Tor network) a sufficiently small number of Sybil identities can present a serious breach of guarantees in the system. Theoretically, the compromising of just two nodes on a circuit is sufficient to identify the sender and the receiver of the communication on such a network [24]. On the other hand, the compromise of a sufficiently large number of identities in the network would enable the attacker to monitor an arbitrary number of circuits. Other applications where the number of identities matters include file sharing systems [25], among many others.

To sum up, the attacker's objective is to maximize the number of Sybil identities in the overlay, although in a few systems, a small number of Sybil identities suffice to thwart the application.

Defense Goals and Success Metric

The ultimate objective of Sybil defenses is to eliminate the Sybil attack by detecting Sybil identities, or peers that generate such identities, and isolate them from the overlay. However, this ultimate goal is not always possible due to fact that most defenses, except in the centralized trusted certification–based scheme, have false positives and false negatives in their detection mechanism that might tolerate some Sybil nodes (false negatives) while mistakenly reporting other honest nodes as false positives. With false positives, honest nodes are reported to be Sybil nodes. On the other hand, in false negatives, Sybil nodes are reported to be honest nodes. The realistic and practically attainable goal of defense mechanisms is to minimize the false negative and false positive rates as much as possible.

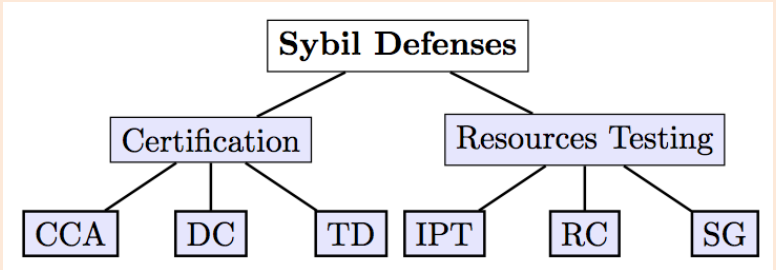

Figure 1. An illustration of the different types of defenses against Sybil attacks in P2P overlays. CCA stands for centralized certification authority, DC for decentralized cryptographic, T/D for trusted devices, IPT for IP testing, R/C for recurring cost, and SG stands for social graph–based approaches

Functional Classification of Sybil Defenses

Besides the broad classification of Sybil defenses into centralized and decentralized defenses, as described in Section 1, the defenses we survey in this paper include two major and broad categories of techniques: trusted certification and resource testing categories, as shown in Figure 1. Among the schemes in the trusted certification category, we survey works that use

a centralized certification authority (CCA), decentralized cryptographic primitives, or trusted devices. Among works that use resource testing, we are particularly interested in works that use IP testing, cost recurrence, and social graphs. While a broad survey is provided for these techniques, detailed descriptions are provided for cryptographic primitives and social graph–based techniques.

Defenses Using Trusted Certification

The trusted certification approach is arguably the most popular method in the context of this study, since Douceur and Donath [6] proved its potential to eliminate the Sybil attack [26]. In the conventional form of this approach, a centralized authority ensures that the identities assigned to each peer are unique and legitimate by matching these identities to pre-assigned credentials. These credentials may include cryptographic keys, synchronized random strings that are usually generated by one-time password generators, or digital certificates issued by the centralized authority.

While the aforementioned conventional form of centralized certification authority is well defined in the literature, there have been some efforts to define distributed certification schemes by applying cryptographic primitives suitable for distributed multiparty models, which enable collaboration among supposedly honest peers to certify other peers joining the overlay. In this section, we review both approaches. In particular, we report on some of the work in the literature on centralized certification authorities and provide details for cryptographic primitives to enable distributed certification systems.

■ Centralized Certification Authority

Centralized trusted certification is potentially the only method that can eliminate the Sybil attack [6]. There has been some work on the usage of centralized certification authorities for credential generation, assignment, and verification in the context of P2P overlays. For instance, works that use social graphs, which are explained in Section 5 and utilize public key cryptography as a building block, assume the authenticity of public keys of users via certificates assigned to users via a centralized authority in an offline phase [19]. There are several examples of schemes that utilize the CCA-based approach [8][9][6][10].

■ Cryptographic Primitives

Some work with cryptographic primitives has been done [11-15]. These primitives aim at providing an infrastructure for authenticating peers in order to make the Sybil attack harder to apply by having only legitimate peers participate in the overlay. Generally, this work tries to exploit a public key infrastructure in a distributed manner using threshold cryptographic ingredients (e.g., secret sharing and threshold signatures) in order to ensure collaboration among supposedly honest users to authenticate peers that join the overlay over its operation time. Interestingly, the explicitly stated motivation beyond some of these primitives [11-14] is that many of the non-cryptographic protocols in the literature assume the existence of a certification system for legitimate users in the overlay (e.g., SybilGuard and SybilLimit). Hence, the cryptographic approaches are designed to ensure successful operation of such protocols. In this section, we review two cryptographic approaches—variations of the work by Lesueur et al. [11] that have appeared in other research [12-14].

■ Trusted Devices

Similar to the idea of trusted certification, some research suggested the usage of trusted devices or trusted modules that store certificates, keys, or authentication strings previously assigned to users by a centralized authority. Such devices are hard to obtain because of their potentially high price, and hence can be used to limit opportunities for Sybil attacks. Examples of such mechanisms are proposed by Rodrigues et al. [27] and Newsome et al. [28], although the latter work is on wireless sensor networks. In theory, when the intent of the attacker is known in advance, these defenses might be effective. However, in cases such as anonymity (Tor, for instance) and recommender systems, given that fewer Sybil identities can cause great harm, these defenses are obsolete.

Defenses Using Resource Testing

The basic idea beyond the resource testing approach for defending against a Sybil attack is to find out if a set of identities associated with supposedly different users own enough resources that match the number of identities. These resources may include computation power, bandwidth, memory, IP address, or even trust credentials. Though the ineffectiveness of the

idea of resource testing was demonstrated by Douceur and Donath [6], some researchers have argued that it can be a minimal defense (i.e., the method is not supposed to entirely eliminate the attack, but make it harder to apply).

In theory, the majority of schemes in this category of defense limit the number of Sybil identities to a smaller number than in the scenario without any defense in place. However, in practice, even a smaller number of Sybil identities is enough to thwart the availability and security of many systems. For instance, as mentioned before, anonymizer systems such as Tor depend on two nodes per circuit. Also, it is enough to have fake identities totaling just 1% of users in online reputation systems in order to out-vote legitimate nodes. In spite of that, we review some of the work that used the resource testing approach and demonstrate their shortcomings, although we should keep in mind that these schemes are mitigating the attack (i.e., discouraging the attacker) rather than eliminating it.

IP Testing

Generic testing schemes include testing the IP address of peers, trying to determine their locations and matching them to their activities. In particular, if a certain amount of activity is generated from the same geographic area, it is likely that some of this activity is due to Sybil identities. Besides, the assumption in such works is that it is not cheap to obtain IP addresses in different geographic areas. For example, Freedman and Morris [29] introduced Tarzan, in which IP addresses of peers are tested based on their geographic location in a particular autonomous system. Similar results were introduced by Cornelli et al. [30].

The main assumption in this work is that IP addresses are hard to obtain in geographically wide areas. However, with recent indicators for the existence of gigantic botnets [31] as well as compromised hosts under control of a single administrative entity and residing in different autonomous systems, it is quite certain that such defense mechanisms are useless.

Recurring Cost

Some work has suggested recurring cost as a method of defending against the Sybil attack. In particular, computational puzzles [16, 32] and Turing tests (e.g., CAPTCHA [33]) are suggested as solutions. Other similar practical solutions that are widely used are phone numbers (like Google email verification) or email addresses (as in social network site registrations). However, for the same reason that IP testing would not work against an attacker that controls a botnet, these cost-based schemes will not work as well. Furthermore, for CAPTCHA-like solutions, it has been shown that a Sybil attacker might post the CAPTCHA tests on sites controlled by her for users who may solve the test for getting access to the service provided by the attacker. Also, some versions of CAPTCHA are vulnerable to image processing attacks [34].

Social Network–Based Sybil Defenses

While most of the previously proposed solutions to a Sybil attack in distributed systems have limitations and shortcomings in one way or another, social network–based Sybil defenses try to overcome such shortcomings in several elegant ways. First, social network–based Sybil defenses are mostly decentralized solutions to a Sybil attack, which means these designs operate without any centralized authority—a feature that is highly desirable and necessary in most distributed systems. This decentralized model is further made easier thanks to the random walk theory, an ingredient mostly utilized in these defenses. Second, these defenses utilize trust of the social links among social nodes, making collaboration among honest nodes possible and easy. Third and last, these defenses were shown in several studies [18-23] to be practical and effective in defending against Sybil attacks at low cost and are further developed as components in many services, including distributed hash tables (DHTs) and Sybil-resistant voting, and are utilized in mobile network routing.

Although they differ greatly in their design details and operation, all social network–based Sybil defenses have two common assumptions: an algorithmic property (called the fast mixing property) and trust. First, these defenses are based on the fast mixing property of social graphs (a property that we formally define below). Informally, the fast mixing property of the social graph implies that the “honest” nodes in such a graph are well enmeshed, and the honest region does not contain a sparse-cut (a cut that connects two large subsets of honest nodes with a few social links). For simplicity, the fast mixing property of social graphs implies that a random walk from any arbitrary node in the social graph would reach very close to the stationary distribution of the Markov chain defined on that graph after a few steps. The number of steps is suggested to be 10 to 15 in a network of a million nodes.

The second assumption common to this vein of defense is trust. In particular, all of these defenses assume a high trust value in the underlying social graphs, as indicated, for example, by face-to-face interactions among the nodes. This particular assumption is necessary in order to determine the difficulty of infiltrating the social network by arbitrarily establishing many attacker social links. While the operation of a Sybil defense to correctly identify “honest” nodes in the

social graph is guaranteed by the fast mixing assumption, and the construction of the corresponding scheme that uses such an algorithmic property, the power to identify Sybil nodes is only guaranteed assuming that the attacker (or attackers' collectivity) control(s) a few links amongst themselves and other honest nodes in the social graph. (Such links are called attack edges.)

SybilGuard

The SybilGuard design from Yu et al. [19, 20] uses the fast mixing property of trust-possessing social networks to detect Sybil nodes. Technically, SybilGuard consists of two phases: initialization and online detection. In the initialization phase, each node constructs its routing table as a random permutation of its adjacent nodes for pairs of incoming and outgoing edges. Next, each node initiates a random walk of length w = O(n×log n) and propagates it to its adjacent nodes following the routing tables constructed using the random permutation. Each node on the path of the random walk publicly registers the random walk originator and later acts as a witness for that node if that node becomes suspect. Furthermore, using back-traceability of the random walks, each originator of a random walk receives the list of "witnesses" (i.e., the nodes that registered the originator's public key and that lie on the path constructed by the random walk of the originator).

In the online phase, a verifier determines whether a suspect is honest or not. First, the suspect sends the addresses of the witnesses on its random route to the verifier. Accordingly, the verifier compares the list of witness to its list of the verifier route. If there is no intersection between the two sets (an event with a very low probability) the verifier aborts and rejects the node. Otherwise, the verifier continues by contacting the nodes on the intersection between the two sets to verify if the suspect has a public key registered with them. If the intersection nodes verify the suspect, the verifier accepts the node; otherwise, it marks it as a Sybil node.

SybilLimit

Unlike SybilGuard in which a single long random walk is used, SybilLimit [18] suggests the use of several shorter random walks. Also, unlike SybilGuard where public keys of verifiers and suspects are registered on nodes in the social graph, SybilLimit suggests the registration of such keys on edges in the social graphs. SybilLimit consists of an initialization phase and an online verification phase. In the initialization phase, each node constructs its routing table using the same method described in SybilGuard and performs r=O(√n) random walks, each of length w=O(log n) where O(log n) is the mixing time of the social graph, which is 10 to 15 in a million-node social graph, and is theoretically assumed to be a sufficient sample from a distribution that is very close to the statistical distribution. Unlike SybilGuard, where all nodes on the path of the random route are used for registering the public key of the originator of the random walk, the last *edge* in each walk among the r random walks is used for registering the public key of that originator node (each such edge is called a *tail*). In particular, the public key of the originator of the walk is registered at the last node in the walk under the last edge through which the random walk arrived. In addition, using the back-traceability property of the random routes, the witnesses that register the public key of the originator node (which could be either a suspect or a verifier) return their identities to that node. Every node in the social graph performs the same process, and collects sets of witnesses (or verification nodes).

In the online phase, as with SybilGuard, the suspect sends the identifiers and addresses of the witnesses to the verifier node, which compares the witnesses in the suspect's list trying to find a match. If a match is found in the two sets by the verifier, it asks a witness with common identity in both sets to verify the identity of the suspect node and decides whether to accept or reject the node based on the outcome of this process. If no intersection happens between the two sets (a very low probability) the verifier aborts and rejects the node, labeling it an attacker.

The main ingredients used for reasoning out the provable guarantees of SybilLimit are same as those in SybilGuard. In particular, given that the random walk length w is the mixing time of the social graph, the last node selected in such a random walk is found according to the stationary distribution. Furthermore, the last edge in the random walk is selected "almost" uniformly at random from the edges in the social graph. In addition, given that r=O(√n), an intersection between the sampled edges of the verifier and the suspect exists with overwhelming probability if the hidden constant r_0 (where r=r_0×√n) is chosen correctly. The authors refer to this condition as the *intersection* condition, which is used to ensure a high probability for intersection of random walks by honest nodes. As in SybilGuard, assuming g attacker edges, the attacker is allowed to register public keys of Sybil identities on at most g×w×r = O(g√n×log n) tails (called *tainted tails*). In such cases, each attached edge introduces additional O(log n) Sybil identities (assuming that the attacker uses the optimal attack strategy by registering different public keys of different Sybil identities with each possible tainted tail).

SybilLimit also greatly depends on w for its security. Since there is no mechanism for estimating the exact value of the parameter, underestimating or overestimating such a parameter are both problematic, as shown above. SybilLimit also provides a "benchmarking technique" for estimating this parameter, which does not provide a guarantee of the quality of the estimation of the parameter. Finally, SybilLimit can provide guarantees on the number of Sybil identities introduced per attack edge as long as g=o(n/log n). Notice that both SybilGuard and SybilLimit do not require global knowledge of the social network they operate on, and can be implemented in a fully decentralized manner.

SybilInfer

SybilInfer [40] uses a probabilistic model defined over random walks (called traces) in order to infer the extent to which a set of nodes, X, which generated such traces, is honest. The basic assumptions in SybilInfer are that each node has a global view and knowledge of the social network, the network is fast mixing, and the node that initiates SybilInfer is an honest node. Technically, SybilInfer tries to ultimately label the various nodes in the graph into honest or Sybil nodes. In SybilInfer, each node in a network of n nodes performs s walks; hence, the overall number of walks in the universal trace is $s \times n$. Each trace among these traces consists of the first node (the initiator of the random walk) and the last node in the random walk (i.e., the tail). Unlike the uniform (over node degree) transition probability used in SybilGuard and SybilLimit, SybilInfer defines the transition matrix uniform over nodes, thus penalizing nodes with a higher degree. The ultimate goal of the operation of SybilInfer is to compute probability P(X=Honest|T), that is, compute the probability that a set of nodes X is honest, given the traces, T. This probability is computed using Bayes' theorem.

SybilInfer also uses non-trivial techniques for sampling the honest configuration that is used initially to determine the honesty of a set of nodes from their traces. This sampling is performed using the Metropolis-Hasting algorithm by initially considering a set X_0 and modifying the set by either removing or adding nodes to the set. Each time, with probability P_{add}, a new node x from X'_0 is added to X_0 to make $X'=X_0 \vee x$, or a node in X_0 is removed with probability P_{remove}. The process is performed for $n \log n$ rounds in order to obtain a good sample independent of X_0.

SumUp

Unlike SybilGuard and SybilLimit, which are generic to the problem of node admission and decentralized (in the sense that they do not require a single node to carry global information about the social graph), and SybilInfer, which is applied to infer the honesty of nodes, SumUp [35] tries to tackle a Sybil attack in the context of vote aggregation. In this context, a node—called the vote collector—wishes to collect votes in a Sybil-resistant manner from other nodes in the network. That is, among a given number of votes on an object, the vote collector wishes to increase the fraction of votes accepted from honest nodes, reduce the fraction of accepted votes cast by the attacker through his attack edge, and identify attackers once they misbehave repeatedly. At the core of SumUp, a link capacity assignment mechanism is used to adaptively assign capacities to links in the trust-possessing social graph, restricting the amount of votes propagated to the vote collector from the voter's side. The adaptive vote flow mechanism of SumUp uses two observations of conventional online voting systems: a few users in the system vote on a single object and—if such voting system is implemented on top of a social graph—the congestion is only at links close to the vote collector. Accordingly, SumUp suggests distributing a number of tickets on the different links in the social graph based on their distance (according to some levels, computed using the breadth-first search algorithm) from the vote collector.

One obvious drawback of the technique is its high computational requirements: the running time of a typical algorithm, such as the Ford-Fulkerson algorithm, would require an order of the number of edges of operations for collecting the vote of a single voter. The authors further suggest a heuristic that uses only an order of the graph diameter as the number of steps, where each node greedily selects a node at the higher level through which it is connected using a non-zero capacity, propagating the vote until it reaches the vote collector. At any time step, given that the greedy step may not result in a non-zero capacity, each node is allowed to explore other nodes for paths at the same or lower level.

GateKeeper

GateKeeper [21] borrows tools from both SumUp and SybilLimit for efficient operation. In particular, it tries to improve the performance of SybilLimit by incorporating the ticket distribution component of SumUp. Unlike SumUp, where nodes are admitted through a non-zero path from the voter to the collector, as explained earlier, GateKeeper only considers the ticket distribution phase of SumUp, where tickets are used for admitting nodes via an admission controller. Such tickets are propagated from the controller to all nodes, as with SumUp. However, in order to limit the attacker's chances of receiving more tickets and to reduce his overall advantage, a controller in GateKeeper randomly selects m different random nodes, called "vantage nodes," where a suspect node is admitted if and only if it receives $m \times f_{admit}$ tickets (where f_{admit} is the fraction of randomly selected vantage points; 0.2 is used in GateKeeper) from different vantage points. Therefore, a node is admitted if it is admitted by such a fraction of the vantage points. To combat double spending, GateKeeper suggests the use of cryptographic signature chains of the paths through which the tickets are spent (propagated to the controller).

Other Social Network–Based Defenses and Applications

SPROUT [36] is another DHT routing protocol that uses social links of trust-possessing social graphs to route information to users operating on top of the social networks. SPROUT, in fact, builds on top of Chord [3] by adding additional links

(routing table entries) in Chord to other users in the social network of any given nodes that are online at any one time. By doing so, SPROUT claims to improve reliability and the distribution of the load of Chord itself.

Whanau was originally presented by Lesniewski-Laas [23], whereas Lesniewski-Lass and Kaashoek [22] included further analysis and proof of performance and security as well as implementation and demonstration of end-to-end guarantees.

Danezis et al. [1] used bootstrap graphs (trees that characterize introduction relationships in the DHT), in order to defend against a Sybil attack. By modifying the operation of Chord regarding the DHT of interest such that each node returns addresses of all nodes that it knows (including introduction points), the authors devise several strategies to reduce the impact of a Sybil attack. Unlike the original Chord, which uses closeness over Chord as a metric for routing (query), the solution considers several strategies for routing, including diversity, mixed, and zig-zag. The authors show experimentally that such strategies, when actually under a Sybil attack, can be used to more efficiently perform Sybil-resistant DHT lookups with fewer queries than required by the plain Chord design. The design of MobID [37] is a social-network based Sybil defense that claims to provide a robust defense for mobile environments, whereas existing defenses have largely been designed for peer-to-peer networks and are based on the random walk theory. Furthermore, MobID uses "betweenness" (a graph-theoretic measure in the social graph) to determine the goodness of nodes in order to defend against Sybil attacks. The work, however, does not seem to provide any provable guarantees.

■ Recent Analyses and Supplementary Work

Mohaisen et al. [38] initiated a study of mixing time as the basic assumption used in Sybil defenses and showed negative results on its quality in many social networks. Viswanath et al. conducted an experimental analysis of Sybil defenses based on social networks [39]. Their study aimed at comparing different defenses (namely, SybilGuard [20], SybilLimit [18], SybilInfer [40], and SumUp [35]) independent of the data sets being used, by decompiling the defenses. They showed that the different Sybil defenses work by ranking different nodes based on how well connected these nodes are to a trusted node (the verifier). Also, they show that the different Sybil defenses are sensitive to the community structure in social networks, and community detection algorithms can be used to replace random walk–based Sybil defenses. Mohaisen et al. arrived at a similar insight [41] by showing that the core structure of social graphs is related to mixing time; Mohaisen and Hollenbeck utilized those findings to improve the mixing time of poorly mixed ones [42]. Mohaisen et al. also studied how trust affects the performance of Sybil defenses [43].

Related Work

Related to our work, Levine et al. [26] proposed a broad survey of solutions for Sybil attacks in general settings, including P2P overlays. Unlike our work, they emphasized classifying the literature broadly, rather than defining merits and shortcomings of each class of work. Our survey, however, greatly benefited from their classification, although the set of schemes reviewed in our survey is greatly different. In particular, the main technical content of our survey reviews research that was published after publication of the survey by Levine et al. [26]. Related to social network–based defenses, Yu presented an intriguing tutorial and a survey [44].

Conclusion

The Sybil attack is very powerful when applied to P2P overlays, and countermeasures against it are harder to implement than in other networking settings because of the P2P overlay nature: centralized authorities necessary for security enforcement are discouraged, and sometimes absent, from P2P overlay designs. In this article, we review the literature on different methods used to defend against the Sybil attack in P2P overlays. We show the different defenses' assumptions, features, and shortcomings and compare them to each other.

References

[1] G. Danezis, C. Lesniewski-laas, M. F. Kaashoek, R. Anderson, "Sybil-resistant DHT routing," *Lecture Notes in Computer Science*, pp. 305-318, 2005.

[2] P. Maymounkov, D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," *Lecture Notes in Computer Science*, pp. 53-65, 2002.

[3] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proc. of SIGCOMM*, pp. 149-160, 2001. Article (CrossRef Link)

[4] Y. Wang, X. Yun, Y. Li, "Analyzing the characteristics of Gnutella overlays," in *Proc. of IEE ITNG*, pp. 1095-1100, 2007. Article (CrossRef Link)

[5] V. Pathak, L. Iftode, "Byzantine fault tolerant public key authentication in peer-to-peer systems," *Computer Networks*, vol. 50, no. 4, pp. 579-596, 2006. Article (CrossRef Link)

[6] J. Douceur, J. S. Donath, "The Sybil attack," in *Proc. of ACM IPDPS*, pp. 251-260, 2002.

[7] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, F. Xiao, "Dsybil: Optimal sybil-resistance for recommendation systems," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 283-298, 2009. Article (CrossRef Link)

[8] M. Castro, P. Druschel, A. J. Ganesh, A. I. T. Rowstron, D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *Proc. of USENIX OSDI*, 2002. Article (CrossRef Link)

[9] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, R. Wattenhofer, "Farsite: Feder- ated, available, and reliable storage for an incompletely trusted environment," in *Proc. of USENIX OSDI*, 2002. Article (CrossRef Link)

[10] J. Ledlie, M. I. Seltzer, "Distributed, secure load balancing with skew, heterogeneity and churn," in *Proc. of IEEE INFOCOM*, pp. 1419-1430, 2005.

[11] F. Lesueur, L. Me, V. V. T. Tong, "An efficient distributed PKI for structured P2P networks," in *Proc. of IEEE P2P*, pp. 1-10, 2009. Article (CrossRef Link)

[12] F. Lesueur, L. Me, V. V. T. Tong, "A distributed certification system for structured P2P networks," *Lecture Notes in Computer Science*, vol. 5127, pp. 40-52, 2008. Article (CrossRef Link)

[13] F. Lesueur, L. Me, V. V. T. Tong, "A Sybil-resistant admission control coupling Sybilguard with distributed certification," in *Proc. of WETICE*, Washington, DC, USA, pp. 105-110, 2008. Article (CrossRef Link)

[14] F. Lesueur, L. Me, V. V. T. Tong, "A Sybilproof distributed identity management for P2P networks," in *Proc. of IEEE ISCC*, pp. 246-253, 2008. Article (CrossRef Link)

[15] A. Avramidis, P. Kotzanikolaou, C. Douligeris, "Chord-PKI: Embedding a public key infrastructure into the chord overlay network," in *Proc. of EuroPKI*, pp. 354-361, 2007.

[16] N. Borisov, "Computational puzzles as sybil defenses," in *Proc. of the 6th IEEE Conference* on *Peer-to-Peer Computing*, pp. 171-176, 2006. Article (CrossRef Link)

[17] H. Yu, P. B. Gibbons, M. Kaminsky, "Toward an optimal social network defense against Sybil attacks," in *Proc. of ACM PODC*, pp. 376-377, 2007. Article (CrossRef Link)

[18] H. Yu, P. B. Gibbons, M. Kaminsky, F. Xiao, "SybilLimit: A near- optimal social network defense against Sybil attacks," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 3-17, 2008. Article (CrossRef Link)

[19] H. Yu, M. Kaminsky, P. B. Gibbons, A. Flaxman, "SybilGuard: defending against Sybil attacks via social networks," in *Proc. of ACM SIGCOMM*, pp. 267-278, 2006. Article (CrossRef Link)

[20] H. Yu, M. Kaminsky, P. B. Gibbons, A. D. Flaxman, "Sybilguard: defending against Sybil attacks via social networks," *IEEE/ACM Transaction on Networks*, vol. 16, no. 3, pp. 576-589, 2008. Article (CrossRef Link)

[21] N. Tran, J. Li, L. Subramanian, S. S. Chow, "Optimal sybil-resilient node admission control," in *Proc. of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, 2011.

[22] C. Lesniewski-Lass, M. F. Kaashoek, "Wh anau: A sybil-proof distributed hash table," in *Proc. of the 7th USENIX Symposium on Network Design and Implementation*, pp. 3-17, 2010.

[23] C. Lesniewski-Laas, "A Sybil-proof one-hop DHT," in *Proc. of the 1st Workshop on Social Network Systems*, pp. 19-24, 2008. Article (CrossRef Link)

[24] P. F. Syverson, D. M. Goldschlag, M. G. Reed, "Anonymous connections and onion routing," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 44-54, 1997.

[25] P. Wang, J. Tyra, E. Chan-tin, T. Malchow, D. F. Kune, Y. Kim, "Attacking the Kad network," in Proc. of the 4th International Conference on Security and Privacy in Communication Networks, 2009. Article (CrossRef Link)

[26] B. Levine, C. Shields, N. Margolin, "A survey of solutions to the Sybil attack," Technical report, University of Massachusetts Amherst, Amherst, MA, 2006.

[27] R. Rodrigues, B. Liskov, L. Shrira, "The design of a robust peer-to-peer system," in *Proc. of the10th ACM SIGOPS European Workshop*, pp. 1-10, 2002.

[28] J. Newsome, E. Shi, D. Song, A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proc. of ACM IPSN*, pp. 259-268, 2004. Article (CrossRef Link)

[29] M. J. Freedman, R. Morris, "Tarzan: a peer-to-peer anonymizing network layer," in *Proc. of ACM CCS*, pp. 193-206, 2002. Article (CrossRef Link)

[30] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, "Choosing reputable servants in a P2P network," in *Proc. of ACM WWW*, pp. 376- 386, 2002.

[31] B. B. Kang, E. Chan-Tin, C. P. Lee, J. Tyra, H. J. Kang, C. Nunnery, Z. Wadler, G. Sinclair, N. Hopper, D. Dagon, Y. Kim, "Towards complete node enumeration in a peer-to-peer botnet," in Proc. of ACM ASIACCS, pp. 23-34, 2009. Article (CrossRef Link)

[32] F. Li, P. Mittal, M. Caesar, N. Borisov, "SybilControl: practical Sybil defense with computational puzzles," in *Proc. of the 7th ACM workshop on Scalable trusted computing*, pp. 67-78, 2012. Article (CrossRef Link)

[33] L. von Ahn, M. Blum, N. J. Hopper, J. Langford, "CAPTCHA: Using hard AI problems for security," Lecture Notes in Computer Science, vol. 2656, pp. 294-311, 2003.

[34] J. Yan, A. S. E. Ahmad, "Breaking visual captchas with naive pattern recognition algorithms," in *Proc. of IEEE ACSAC*, Washington, DC, USA, pp. 279-291, 2007. Article (CrossRef Link)

[35] N. Tran, B. Min, J. Li, L. Subramanian, "Sybil-resilient online content voting," in *Proc. of USENIX NSDI*, Berkeley, CA, USA, 2009.

[36] S. Marti, P. Ganesan, H. Garcia-Molina, "DHT routing using social links," in *Proc. of IEEE IPTPS*, pp. 100-111, 2004. Article (CrossRef Link)

[37] D. Quercia, S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in *Proc. of IEEE INFOCOM*, pp. 336-340, 2010. Article (CrossRef Link)

[38] A. Mohaisen, A. Yun, Y. Kim, "Measuring the mixing time of social graphs," in *Proc. of ACM IMC*, pp. 383-389, 2010. Article (CrossRef Link)

[39] B. Viswanath, A. Post, K. P. Gummadi, A. Mislove, "An analysis of social network-based Sybil defenses," in *Proc. of ACM SIGCOMM*, 2010. Article (CrossRef Link)

[40] G. Danezis, P. Mittal, "SybilInfer: Detecting Sybil nodes using social networks," in *Proc. of ISOC NDSS*, 2009.

[41] A. Mohaisen, H. Tran, N. Hopper, Y. Kim, "Understanding social networks properties for trustworthy computing," in *Proc. of IEEE ICDCS Workshops*, pp. 154-159, 2011. Article (CrossRef Link)

[42] A. Mohaisen, S. Hollenbeck, "Improving social network-based Sybil defenses by augmenting social graphs," in *Proc. of WISA*, 2013.

[43] A. Mohaisen, N. Hopper, Y. Kim, "Keep your friends close: Incorporating trust into social network-based Sybil defenses," in *Proc. of IEEE INFOCOM*, pp. 1943-1951, 2011.

[44] H. Yu, "Sybil defenses via social networks: a tutorial and survey," *ACM SIGACT News*, vol. 42, no. 3, pp. 80-101, 2011. Article (CrossRef Link)

Aziz Mohaisen is a research scientist at VeriSign Labs. His research interests are broadly focused on security, privacy, measurement, and analysis of complex and emerging network systems. His recent work emphasized data-driven security and its application in malware analysis, network routing, information sharing, and Internet-scale reputation. He obtained his PhD in computer science from the University of Minnesota in 2012, where he wrote his dissertation on trustworthy social computing systems.

Joongheon Kim received BS and MS degrees from Korea University, Seoul, Republic of Korea, in 2004 and 2006, respectively. He was with LG Electronics in Seoul from 2006 to 2009. He is now a PhD student at the University of Southern California, Los Angeles. His current research interests are multi-gigabit millimeter-wave wireless systems and device-to-device distributed streaming platforms.